Comparison of online motion planning algorithm for space agent: Artificial Potential Field (APF) and Model Predictive Control (MPC)

Nascivera Davide, Fici Lorenzo

November 29, 2024

Contents

Abstract

This project evaluates various control strategies for robotic systems dedicated to space debris removal, specifically comparing Model Predictive Control (MPC) with controllers integrated with artificial Potential Field (APF) methods. Utilizing simulations, we assess these controllers based on key performance metrics such as computational complexity and efficiency. By building on existing models and comprehensive literature, the study aims to identify the most effective control approach for different operational scenarios, ranging from complex environments with numerous obstacles to simpler tasks in simplified environments. The findings of this research will guide the selection of optimal control strategies, thereby enhancing the reliability and efficiency of space debris mitigation missions.

1 Introduction

Navigating autonomous robots in dynamic and obstacle-rich environments is a fundamental challenge in robotics. Two prominent approaches to address this challenge are the **Artificial Potential Field (APF)** method and **Model Predictive Control (MPC)**. This document provides a mathematical description of both methods as implemented in respective Python scripts, highlighting their underlying principles, system dynamics, optimization strategies, and performance metrics.

2 Theoretical framework

This project will investigate two primary control techniques: Model Predictive Control (MPC), and Artificial Potential Field controller. For the MPC approach, we will adopt the methodology presented in the EL2700 course offered by the Royal Institute of Technology (KTH) which is based on the contents of Borrelli's book and Cannon's lectures notes [4] [5], utilising the associated laboratory project files and lecture notes. These resources will be further supplemented with relevant literature from existing research to provide a comprehensive understanding of MPC for spatial rendezvous such as Hartley's work [8].

Regarding Artificial Potential Field method, we will reference seminal papers and authoritative university textbooks to establish a robust theoretical foundation. For instance, the paper by Chen et al. [6] presents a trajectory planning and control framework designed for spacecraft navigating through dynamic debris swarms. The authors integrate receding horizon control (RHC) and quadratic programming (QP) techniques to ensure real-time trajectory adjustments. In contrast, our research plan aims to evaluate and compare control strategies such as MPC and APF controllers. Still on the same topic we reference the work described in the book of Hu, Shao and Guo [14] which provides an overview of the Artificial Potential Function method and its applications in spacecraft control and motion planning. Instead of focusing solely on trajectory planning, our study emphasizes a comparison of these control techniques in terms of computational complexity, energy efficiency, and accuracy for space debris removal.

We are going to use the work of Assirelli [3], which describes a tutorial on MPC control for a differential drive unicycle (a simple vehicle that moves forward and rotates), to develop a simplified simulation. It covers solving an optimal control problem (OCP) to drive the unicycle from a start to a target configuration. The script includes trajectory optimization, open-loop control, and MPC with a disturbance simulation. The goal is to demonstrate how MPC can help in correcting the robot's trajectory when faced with unexpected events.

The implementation of these control techniques will be developed in-house, with the assistance of generative tools such as GitHub Copilot and similar alternatives to debug and enhance coding efficiency/accuracy. In cases where we incorporate code or projects developed by other researchers, proper citations will be included in the references section to acknowledge their contributions and maintain academic integrity.

2.1 Unicycle Model Dynamics

We decided to model our spacecraft as a unicycle. This is a big simplification for computations that can be justified by comparing the model of the unicycle with the model provided in [18]. The first simplification is moving from a 3D simulation to a 2D one, the second one is removing the equation regarding the mass. So at the end, our model will consist of 3 equations: two for the position along x-axis and y-axis, one for the angle, necessary for the model of the unicycle since we are not having vectorial variables but only scalar ones.

2.1.1 State and Control Vectors

Both APF and MPC approaches utilize the unicycle model to represent the robot's kinematics. The unicycle model is defined by its state and control inputs.

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

where:

- x and y are the Cartesian coordinates of the robot.
- θ is the orientation angle with respect to the positive x-axis.
- v is the linear velocity.
- ω is the angular velocity.

2.1.2 Dynamics Equations

The unicycle model updates its state based on the control inputs as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \begin{bmatrix} v_k \cos(\theta_k)\Delta t \\ v_k \sin(\theta_k)\Delta t \\ \omega_k \Delta t \end{bmatrix}$$

where:
- Δt is the time step.
- k denotes the current time step.

2.2 Artificial Potential Field (APF) Approach

The APF method treats the navigation problem as a potential field problem where the target exerts an attractive potential, and obstacles exert repulsive potentials. The robot moves under the influence of the resultant force derived from these potentials.

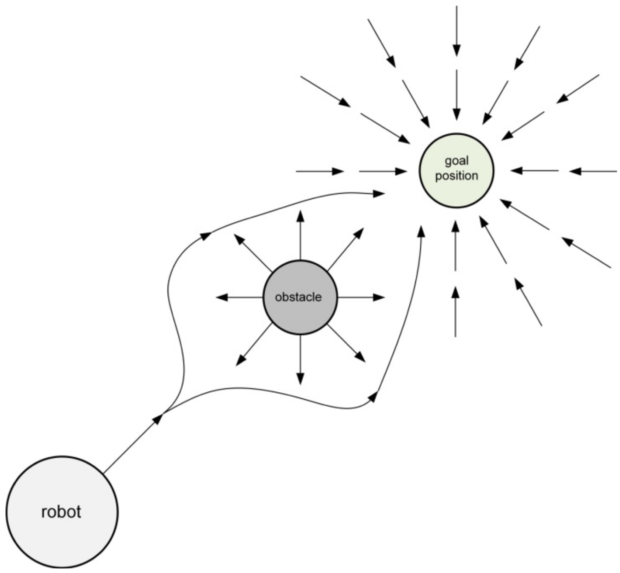


Figure 1: Artificial potential field illustrating goal attraction and obstacle repulsion in robotic path planning. [10].

The attractive potential U_{att} draws the robot toward the target.

$$U_{\text{att}}(\mathbf{p}) = \frac{1}{2} k_{\text{att}} \|\mathbf{p} - \mathbf{p}_{\text{target}}\|^2$$

where:
- $\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}$ is the current position of the robot.
- $\mathbf{p}_{\text{target}} = \begin{bmatrix} x_{\text{target}} \\ y_{\text{target}} \end{bmatrix}$ is the target position.
- k_{att} is the attractive potential gain.

The **attractive force** $\mathbf{F}_{\text{att}}$ is the negative gradient of the attractive potential.

$$\mathbf{F}_{\text{att}} = -\nabla U_{\text{att}} = -k_{\text{att}}(\mathbf{p} - \mathbf{p}_{\text{target}})$$

2.2.1 Repulsive Potential

The **repulsive potential** U_{rep} pushes the robot away from obstacles within an influence radius.

$$U_{\text{rep}}(\mathbf{p}) = \begin{cases} \frac{1}{2} k_{\text{rep}} \left(\frac{1}{d(\mathbf{p})} - \frac{1}{d_0} \right)^2 & \text{if } d(\mathbf{p}) \leq d_0 \\ 0 & \text{otherwise} \end{cases}$$

where:
- $d(\mathbf{p})$ is the distance between the robot and an obstacle.
- d_0 is the influence radius.
- k_{rep} is the repulsive potential gain.

The **repulsive force** $\mathbf{F}_{\text{rep}}$ is the negative gradient of the repulsive potential.

$$\mathbf{F}_{\text{rep}} = \begin{cases} k_{\text{rep}} \left(\frac{1}{d(\mathbf{p})} - \frac{1}{d_0} \right) \frac{1}{d(\mathbf{p})^2} \frac{\mathbf{p} - \mathbf{p}_{\text{obs}}}{d(\mathbf{p})} & \text{if } d(\mathbf{p}) \leq d_0 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

2.2.2 Total Potential and Force

The total potential U_{total} and the resultant force $\mathbf{F}_{\text{total}}$ acting on the robot are given by:

$$U_{\text{total}} = U_{\text{att}} + \sum_{j=1}^{M} U_{\text{rep}_j}$$

$$\mathbf{F}_{\text{total}} = \mathbf{F}_{\text{att}} + \sum_{j=1}^{M} \mathbf{F}_{\text{rep}_j}$$

where M is the number of obstacles.

2.2.3 Control Inputs Based on Potential Forces

The control inputs v and ω are derived from the resultant force $\mathbf{F}_{\text{total}}$:

$$\theta_{\text{desired}} = \arctan 2(F_{\text{total},y}, F_{\text{total},x})$$

$$v = v_{\max} \cdot \cos(\theta_{\text{desired}} - \theta)$$

$$\omega = \omega_{\max} \cdot \text{normalize}(\theta_{\text{desired}} - \theta)$$

Constraints:

- v is clipped to $[-v_{\max}, v_{\max}]$.
- ω is clipped to $[-\omega_{\max}, \omega_{\max}]$.

Normalization Function:

$$\text{normalize}(\alpha) = (\alpha + \pi) \mod (2\pi) - \pi$$

2.3 Model Predictive Control (MPC) Approach

Model Predictive Control (MPC) is an advanced control strategy that involves solving an optimization problem at each time step to determine a sequence of future control actions. These actions are optimized over a finite prediction horizon to minimize a defined cost function, typically aiming to balance path accuracy, control effort, and obstacle avoidance. Unlike Artificial Potential Field (APF), MPC explicitly models system dynamics and predicts future states, allowing it to account for constraints such as speed limits, obstacle boundaries, and control input limits. [19, 15].

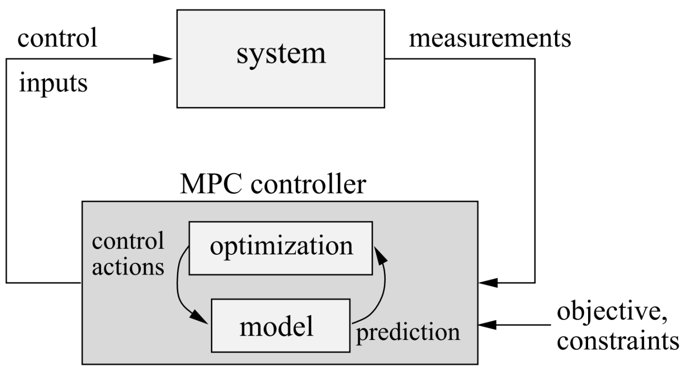


Figure 2: Model Predictive Control (MPC) framework illustrating the optimization process over a prediction horizon. Adapted from [2].

2.3.1 Optimization Problem for MPC

The optimization problem for Model Predictive Control (MPC) involves finding the optimal sequence of control inputs $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}$ that minimizes the cost function J subject to system dynamics and control constraints.

$$\begin{aligned} \min_{\mathbf{u}_1,\ldots,\mathbf{u}_N} \quad & J(\mathbf{u}_1, \ldots, \mathbf{u}_N) \\ \text{subject to} \quad & \mathbf{x}_{i+1} = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i), \quad \forall i = 1, \ldots, N \\ & v_{\min} \le v_i \le v_{\max}, \quad \forall i = 1, \ldots, N \\ & \omega_{\min} \le \omega_i \le \omega_{\max}, \quad \forall i = 1, \ldots, N \end{aligned}$$

2.3.2 Cost Function Components

The cost function J in Model Predictive Control (MPC) is designed to balance multiple objectives, ensuring optimal system performance. It comprises the following components:

- **State Error Cost:**
$$J_{\text{state}} = \sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{x}_{\text{target}})^T \mathbf{Q} (\mathbf{x}_i - \mathbf{x}_{\text{target}})$$
This term penalizes deviations of the system state from the desired target state, promoting accurate trajectory tracking [19].
- **Control Effort Cost:**
$$J_{\text{control}} = \sum_{i=1}^{N} \mathbf{u}_i^T \mathbf{R} \mathbf{u}_i$$
This component penalizes excessive control inputs, encouraging energy-efficient and smooth control actions [19].
- **Obstacle Avoidance Cost:**
$$J_{\text{obs}} = \sum_{i=1}^{N} \sum_{j=1}^{M} w_{\text{obs}} e^{-\alpha(d_{ij} - r_{\text{obs}})}$$
Incorporating obstacle avoidance into the cost function ensures the system maintains safe distances from obstacles [17].
- **Terminal Cost:**
$$J_{\text{terminal}} = (\mathbf{x}_{N+1} - \mathbf{x}_{\text{target}})^T \mathbf{Q}_{\text{terminal}} (\mathbf{x}_{N+1} - \mathbf{x}_{\text{target}})$$
This term penalizes the deviation of the final predicted state from the target state at the end of the prediction horizon, enhancing stability and convergence [20].

where:

- $\mathbf{x}_i$ is the predicted state at time step i.
- $\mathbf{x}_{\text{target}}$ is the target state.
- $\mathbf{Q}$ is the state error weight matrix.
- $\mathbf{R}$ is the control effort weight matrix.
- $\mathbf{Q}_{\text{terminal}}$ is the terminal state error weight matrix.
- w_{obs} is the obstacle avoidance weight.
- α is the exponential penalty factor for obstacle avoidance.
- d_{ij} is the distance between the robot at prediction step i and obstacle j.
- r_{obs} is the obstacle radius.
- M is the number of obstacles.

2.3.3 Predicted Trajectory Simulation

Given an initial state $\mathbf{x}_0$ and a sequence of control inputs $\{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$, the predicted trajectory $\{\mathbf{x}_1, \ldots, \mathbf{x}_{N+1}\}$ is simulated using the unicycle dynamics:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \begin{bmatrix} v_k \cos(\theta_k)\Delta t \\ v_k \sin(\theta_k)\Delta t \\ \omega_k \Delta t \end{bmatrix}, \quad \forall k = 0, \ldots, N$$

2.3.4 Control Input Application and Initialization

At each iteration k:

1. Solve the optimization problem to obtain the optimal control sequence $\{\mathbf{u}_1^*, \ldots, \mathbf{u}_N^*\}$.
2. Apply the first control input $\mathbf{u}_1^*$ to the robot.
3. Update the robot's state using the unicycle dynamics.
4. Shift the control sequence for the next iteration:

$$\mathbf{u}_{k+1}^0 = \{\mathbf{u}_2^*, \ldots, \mathbf{u}_N^*, \mathbf{0}\}$$

3 Research Question and Hypotheses

The objective is to compare different control techniques for managing agents in charge of dismantling an unidentified vehicle (space debris). Through simulation, we will evaluate which control method is most effective in ensuring accuracy, efficiency in resource consumption, and speed of execution. The evaluation will focus on key performance metrics such as computational complexity, energy efficiency, trajectory accuracy, and responsiveness. This comparison will identify the best approach for operating in complex scenarios such as space debris removal, where accuracy and resource optimization are crucial to mission success.

We hypothesise that MPC will outperform APF in accuracy, control action, and obstacle avoidance in complex scenarios. At the same time, APF will excel in speed and simplicity where control action is less critical.

4 Proposed Methodology

The method employed in this project will follow an iterative "implement-evaluate-validate-refine" approach. Building upon established research in Model Predictive Control (MPC) for collision avoidance, we will adapt a pre-existing spatial object model developed during the MPC course at KTH. Additionally, we will integrate and modify code examples from online repositories to create a robust simulation environment for our experiments. All relevant materials, including code scripts and simulation models, will be systematically organized and made available through the project index for easy access and collaboration. After implementing both models, we will proceed with the evaluation of performances according to some specific key indices that will be deeply described in the next paragraph. After collecting the data we can effectively compare the two models and we'll validate the hypothesis previously formulated. Finally, we're going to fine-tune our models to plot the desired graphs and tables. This structured methodology will enable comprehensive experimentation and analysis, allowing for a thorough evaluation and comparison of different control strategies for space debris removal.

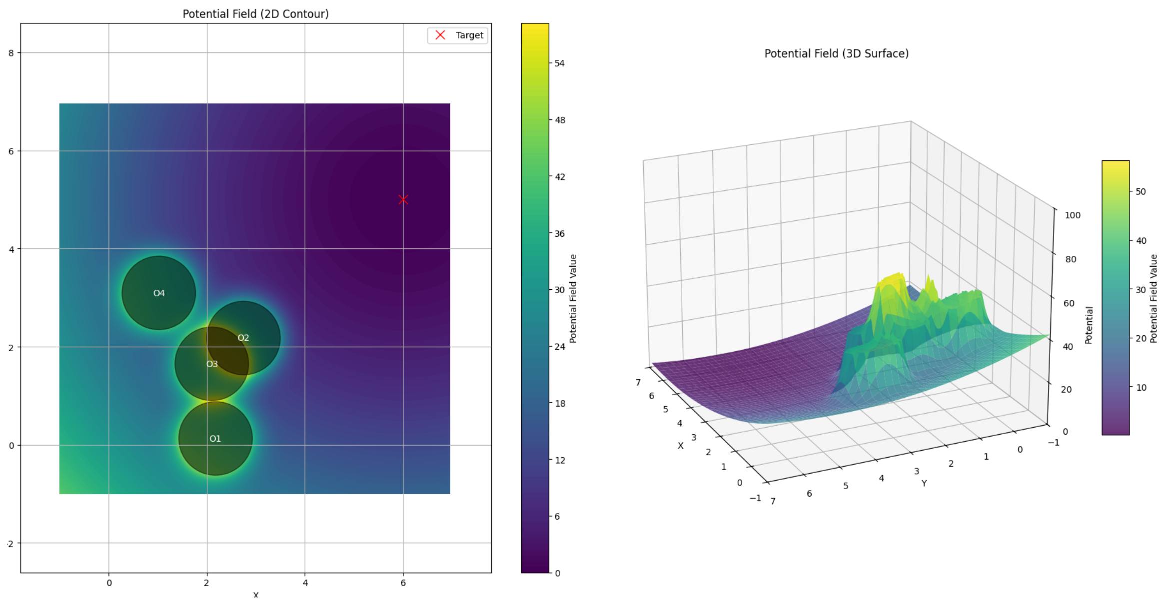


Figure 3: Visualization of the potential field: 2D field with obstacles and goal (left), 3D surface representation (right)

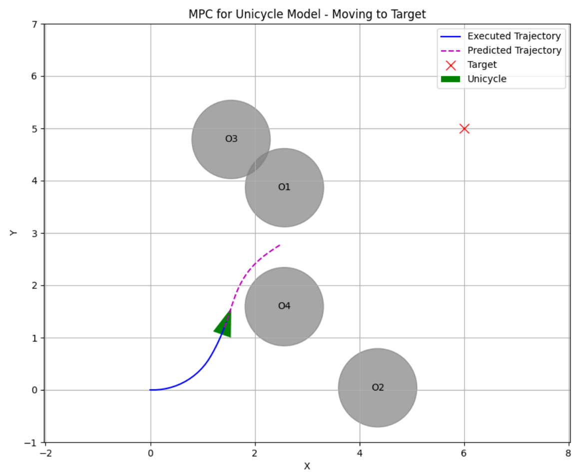


Figure 4: Visualization of the Receding Horizon Model Predictive Control (MPC). The predicted portion is highlighted in purple.

All the different parameters used they can be found at the respective appendix A.

4.1 Metrics Evaluation

To comprehensively assess the performance of path planning algorithms in robotics and autonomous systems, we utilized a set of well-established evaluation metrics. These metrics provide critical insights into the efficiency, safety, accuracy, and computational demands of each algorithm, helping to ensure that the selected approach can navigate reliably and effectively. The following key metrics were used to compare the algorithms in our simulations:

- **Total Path Length**: Measures the distance traveled from the start to the goal, with shorter paths indicating higher efficiency [1].
- **Final Distance to Target**: Assesses the accuracy in reaching the intended goal, where a smaller final distance signifies greater precision [13].
- **Minimum Distance to Obstacles**: Evaluates safety by determining how closely the robot approaches obstacles, with larger distances preferred to avoid collisions [11].
- **Total Control Effort**: Quantifies the cumulative effort required by control inputs to follow the planned path, with lower values indicating energy efficiency and reduced actuator wear [7].
- **Average Computation Time per Iteration**: Measures computational efficiency, essential for real-time applications, with shorter times being more desirable [12].
- **Total Number of Iterations**: Indicates the number of iterations needed for the algorithm to converge to a solution, where fewer iterations suggest faster convergence and computational efficiency [16].
- **L2 Norm of Control Input Changes**: Reflects the smoothness of control inputs by evaluating the magnitude of changes between consecutive inputs. Smaller values denote smoother control actions, beneficial for system stability [9].

These metrics collectively provide a comprehensive evaluation of path planning algorithms, considering factors such as efficiency, accuracy, safety, computational demand, and control smoothness.

5 Results and Analysis

The analysis and corresponding graphs presented in this chapter are derived from the performance metrics discussed in the previous section. This chapter offers a comparative evaluation of two motion planning techniques: Model Predictive Control (MPC) and Artificial Potential Fields (APF). The objective of this analysis is to highlight the relative advantages and limitations of each method within the context of motion planning applications.

This section is divided into two parts. The first part presents results for a specific map, while the second part compares data obtained from simulations of 20 different maps with 4 obstacles generated in random positions.

5.1 Specific Map Analysis

The animations showcasing the trajectory simulations for both MPC and APF approaches are available in a dedicated GitHub repository here. These GIFs provide a dynamic view of the performance metrics discussed in this section.

Metric	**MPC**	**APF**
Map Number	15	15
Total Path Length	7.92	8.98
Final Distance to Target	0.089	0.096
Minimum Distance to Obstacles	0.190	0.441
Total Control Effort	83.81	110.68
Average Computation Time per Iteration	0.159	5.36×10^{-5}
Total Number of Iterations	81	125
Total Number of Function Evaluations	20,282	201
L_2 Norm of Control Input Changes	0.521	2.943
Success	True	True

Table 1: Comparison of Metrics for MPC and APF on Map 15

In Figure 5, we observe that both MPC and APF successfully navigate through Map 15 to reach the target. However, the MPC method follows a more direct and efficient path compared to the APF method. This is further supported by the performance metrics presented in Table 1, where MPC demonstrates a shorter total path length and lower total control effort.

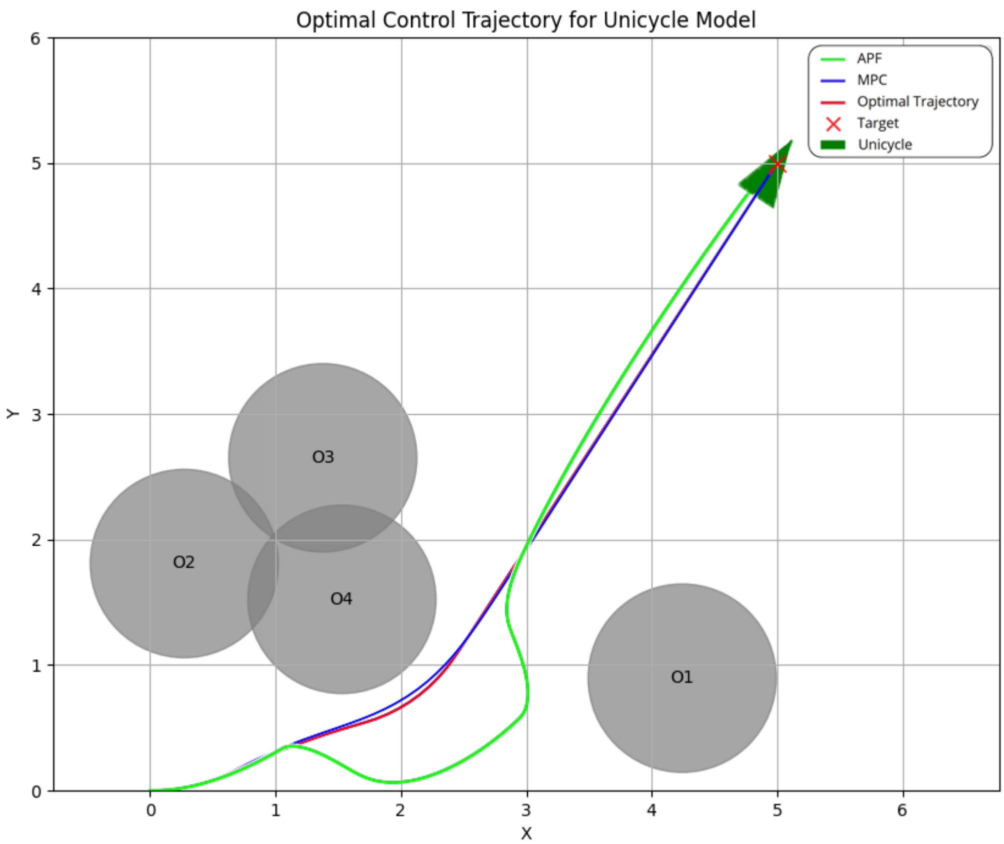


Figure 5: Comparison of MPC and APF on Map 15 where both methods successfully reach the target.

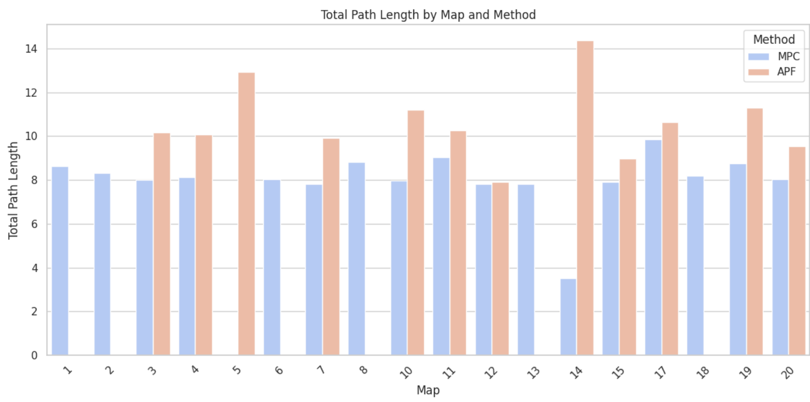


Figure 6: Comparison of Total Path Length for MPC and APF over 20 random maps.

5.2 General Analysis

5.2.1 Data Overview

The datasets used for this analysis were derived from simulated experiments where each method was evaluated across various maps. Essential metrics such as *Total Path Length*, *Final Distance to Target*, *Minimum Distance to Obstacles*, *Total Control Effort*, and *Computation Time* were collected for both MPC and APF. Rows with missing values were removed to ensure data integrity.

5.2.2 Performance Comparison

MPC demonstrated a significantly shorter *Total Path Length* (mean = 8.04) compared to APF (mean = 11.07), indicating a more efficient path to the target. A t-test confirmed this difference as statistically significant ($p < 0.001$) (see Figure 6).

Both methods showed varying final distances to the target, with MPC achieving a lower average final distance (0.35) compared to APF (2.28). This difference suggests that MPC reached closer proximity to the target, making it a preferable choice for precision tasks.

The *Total Control Effort* required for MPC was considerably lower than that for APF, suggesting that MPC required less force and could potentially be more energy-efficient (see Figure 7a).

The *Average Computation Time per Iteration* was substantially lower for APF, with a near-zero mean value, making it more suitable for applications requiring rapid computations (see Figure 7d). However, MPC's higher computation time is offset by its superior accuracy and efficiency.

5.3 Discussion

In conclusion, this analysis highlights the trade-offs between MPC and APF. Results showed that MPC offers smoother control, achieving shorter path lengths, higher success rate for obstacle avoidance, and fewer iterations, although with marginally higher computation per iteration. Conversely, APF performs faster, simpler computations but with less control stability. These insights will guide the selection of control methods based on specific operational needs.

References

[1] "A Review of Path Planning Algorithms". In: *SpringerLink* (2024).

[2] Michele Arnold et al. "Multi-Area Predictive Control for Combined Electricity and Natural Gas Systems". In: *IEEE Transactions on Control Systems Technology* 23.1 (2015), pp. 72–85. DOI: 10.1109/TCST.2014.2318293. URL: https://www.researchgate.net/publication/242397596_Multi-Area_Predictive_Control_for_Combined_Electricity_and_Natural_Gas_Systems.

[3] A. Assirelli. *MPC Control of a Differential Drive Unicycle*. 2023. URL: https://www.miro.ing.unitn.it/wp-content/uploads/2023/03/Main_OCP.html.

[4] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge, UK: Cambridge University Press, 2007. ISBN: 978-1107016880.

[5] John D. Cannon. *Advanced Control Strategies for Collision Avoidance in Spacecraft*. Lecture Notes IFAC World Congress 2008. Royal Institute of Technology (KTH), 2008. URL: https://users.ox.ac.uk/~engs0169/pdf/cannon_ifac08c.pdf.

[6] R. Chen et al. "Trajectory Planning and Control of Spacecraft Avoiding Dynamic Debris Swarm". In: *Aerospace Science and Technology* 151 (2024). DOI: 10.1016/j.ast.2024.109273. URL: https://doi.org/10.1016/j.ast.2024.109273.

[7] "Comparison of Linear and Nonlinear Model Predictive Control in Path Following of Underactuated Unmanned Surface Vehicles". In: *MDPI* (2024).

[8] E. N. Hartley. "A Tutorial on Model Predictive Control for Spacecraft Rendezvous". In: *2015 European Control Conference (ECC)*. 2015, pp. 1355–1361. DOI: 10.1109/ecc.2015.7330727. URL: https://doi.org/10.1109/ecc.2015.7330727.

[9] "Improved Informed-RRT* Based Path Planning and Trajectory Optimization for Mobile Robots". In: *Springer* (2022).

[10] Author(s) Name. "Discrete Artificial Potential Field for Ship Trajectory Planning". In: *Journal of Navigation* (Year). URL: https://www.cambridge.org/core/journals/journal-of-navigation/article/discrete-artificial-potential-field-for-ship-trajectory-planning/32E349D51204DD2BF4CA989E128F8901.

[11] "Path Planning and Trajectory Planning Algorithms: A General Overview". In: *SpringerLink* (2015).

[12] "Path Planning in Complex Environments Using Attention-Based Deep Deterministic Policy Gradient". In: *MDPI* (2024).

[13] "Performance Analysis of Path Planning Techniques for Autonomous Robots". In: *Springer* (2023).

[14] Lei Guo Qinglei Hu Xiaodong Shao. *Intelligent Autonomous Control of Spacecraft with Multiple Constraints*. Springer, 2023, pp. 7–9. DOI: 10.1007/978-981-99-0681-9. URL: https://link.springer.com/book/10.1007/978-981-99-0681-9.

[15] Daniel Raffo. "Nonlinear Model Predictive Control Schemes for Obstacle Avoidance". In: *Journal of Control, Automation and Electrical Systems* 34 (2023), pp. 559–573. URL: https://link.springer.com/article/10.1007/s40313-023-01024-2.

[16] "Sampling-Based Path Planning Algorithms: A Survey". In: *arXiv* (2023).

[17] Marcelo A. Santos, Antonio Ferramosca, and Guilherme V. Raffo. "Nonlinear Model Predictive Control Schemes for Obstacle Avoidance". In: *Journal of Control, Automation and Electrical Systems* 34 (2023), pp. 891–906. URL: https://link.springer.com/article/10.1007/s40313-023-01024-2.

[18] Abolfazl Shirazi and A. H. Mazinan. "Mathematical modeling of spacecraft guidance and control system in 3D space orbit transfer mission". In: *Computational and Applied Mathematics* 35.3 (2016), pp. 865–879. DOI: 10.1007/s40314-015-0290-8.

[19] Shuyou Yu et al. "Model predictive control for autonomous ground vehicles: a review". In: *Autonomous Intelligent Systems* 1.4 (2021). URL: https://link.springer.com/article/10.1007/s43684-021-00005-z.

[20] Alessandro Zanelli, John Lygeros, and Manfred Morari. "Learning Convex Terminal Costs for Complexity Reduction in MPC". In: *Proceedings of the 60th IEEE Conference on Decision and Control (CDC)*. 2021, pp. 1–6. URL: https://cse.lab.imtlucca.it/~bemporad/publications/papers/cdc21_terminal.pdf.

APPENDIX

A Mathematical Formulas

- **Total Path Length**:

$$L = \sum_{k=0}^{K-1} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2$$

- **Final Distance to Target**:

$$d_K = \|\mathbf{x}_K^{(2)} - \mathbf{x}_{\text{target}}^{(2)}\|_2$$

- **Minimum Distance to Obstacles**:

$$d_{\min} = \min_{k,j} \left(\|\mathbf{x}_k^{(2)} - \mathbf{o}_j^{(2)}\|_2 - r_{\text{obs}} \right)$$

- **Total Control Effort**:

$$E = \sum_{k=1}^{K} \|\mathbf{u}_k\|_2^2 = \sum_{k=1}^{K} \left(v_k^2 + \omega_k^2 \right)$$

- **Average Computation Time per Iteration**:

$$\text{Average Time} = \frac{1}{K} \sum_{k=1}^{K} t_k$$

- **Total Number of Iterations**:

$$\text{Total Iterations} = K$$

- **L2 Norm of Control Input Changes**:

$$S = \sum_{k=2}^{K} \|\mathbf{u}_k - \mathbf{u}_{k-1}\|_2^2 = \sum_{k=2}^{K} \left((v_k - v_{k-1})^2 + (\omega_k - \omega_{k-1})^2 \right)$$

B Parameters

The parameters used for the simulation of the Artificial Potential Field (APF) method are listed in Table 2, and those for the Model Predictive Control (MPC) simulation are provided in Table 3.

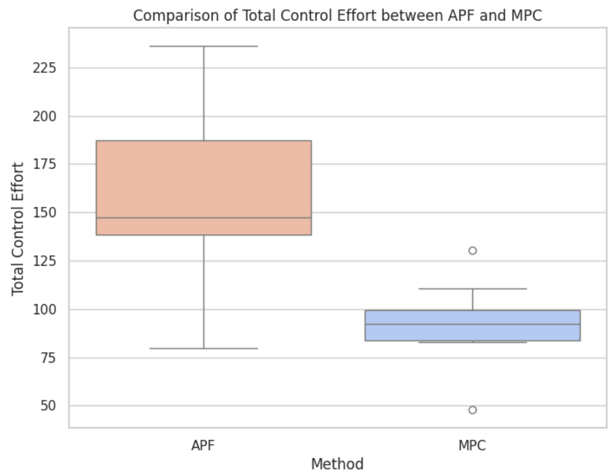
Parameter	**Value**
Random Seed	`randSeed`
Time Step	$dt = 0.1$
Maximum Iterations	`max_iterations` $= 200$
Initial State	`initial_state` $= [0.0, 0.0, 0.0]$
Target Position	`target` $= [6.0, 5.0]$
Obstacle Radius	`obstacle_radius` $= 0.75$
Influence Radius	`influence_radius` $= 1.5$
Linear Velocity Limit	$v_{\min} = -1.0$, $v_{\max} = 1.0$
Angular Velocity Limit	$\omega_{\min} = -\frac{\pi}{4}$, $\omega_{\max} = \frac{\pi}{4}$
Attractive Potential Gain	$k_{\text{att}} = 1.0$
Repulsive Potential Gain	$k_{\text{rep}} = 100.0$
Grid x Range	$x_{\min} = -1$, $x_{\max} = 7$
Grid y Range	$y_{\min} = -1$, $y_{\max} = 7$
Threshold Distance to Target	`threshold_distance` $= 0.15$

Table 2: APF Parameters

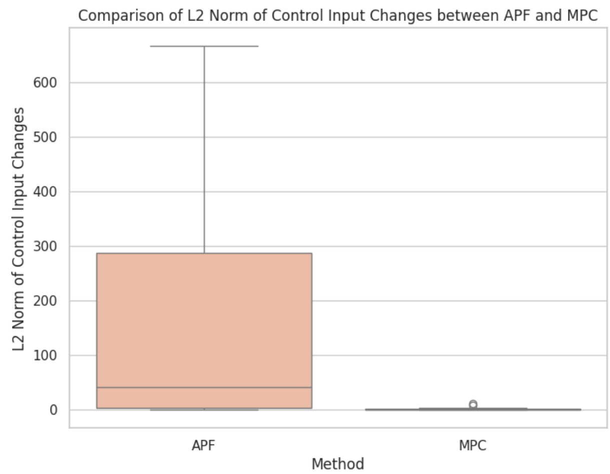
Parameter	Value
Random Seed	`randSeed`
Time Step	$dt = 0.1$
Prediction Horizon	$N = 20$
Initial State	$\texttt{initial\_state} = [0.0, 0.0, 0.0]$
Target State	$\texttt{target} = [6.0, 5.0]$
Obstacle Radius	$\texttt{obstacle\_radius} = 0.75$
Linear Velocity Limits	$v_{\text{min}} = -1.0$, $v_{\text{max}} = 1.0$
Angular Velocity Limits	$\omega_{\text{min}} = -\frac{\pi}{4}$, $\omega_{\text{max}} = \frac{\pi}{4}$
State Error Weight	$Q = 10.0 \times I_2$
Control Effort Weight	$R = 1.0 \times I_2$
Terminal Cost Weight	$Q_{\text{terminal}} = 50.0 \times I_2$
Obstacle Weight	$\texttt{obstacle\_weight} = 100.0$
Exponential Penalty Factor	$\alpha = 10.0$
Grid x Range	$x_{\text{min}} = -1$, $x_{\text{max}} = 7$
Grid y Range	$y_{\text{min}} = -1$, $y_{\text{max}} = 7$

Table 3: MPC Parameters

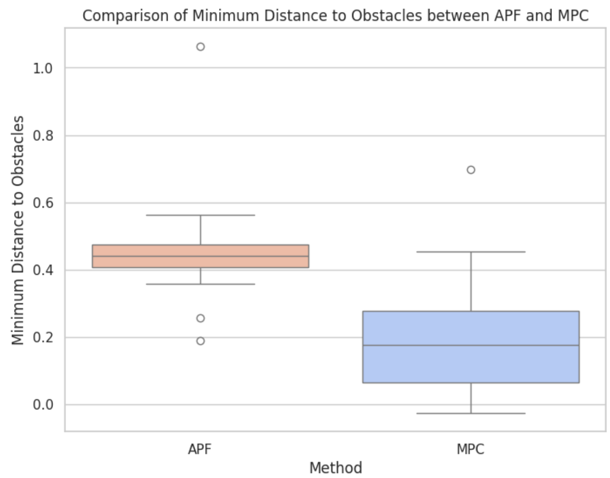
C Additional Graphs



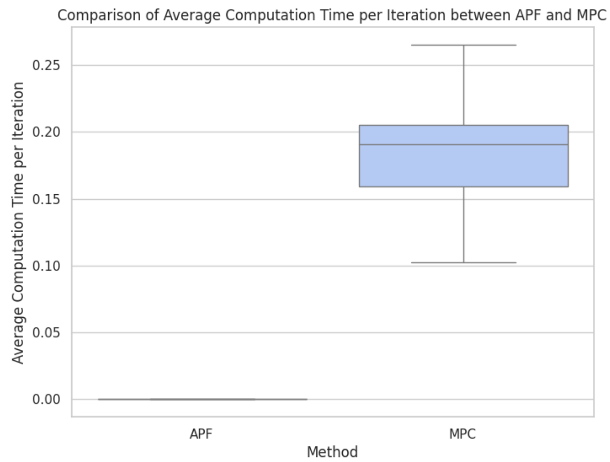
(a) Total Control Effort



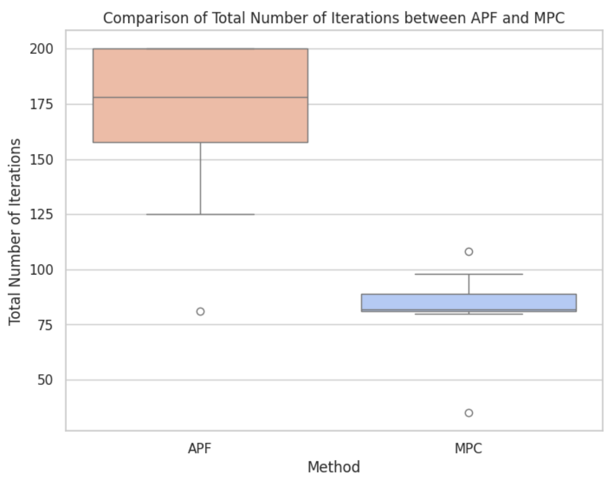
(b) L_2 Norm of Control Input Changes



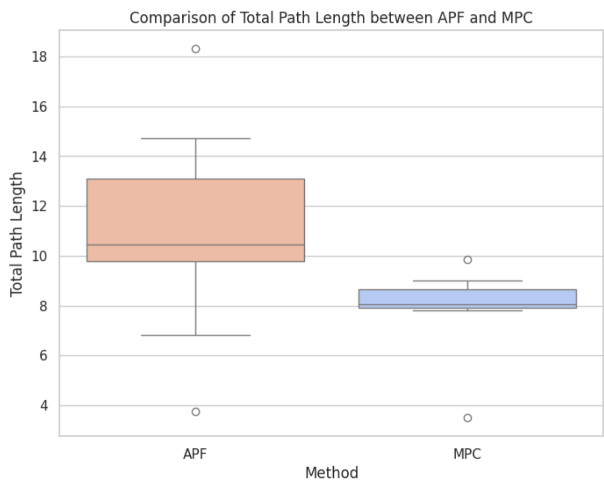
(c) Minimum Distance to Obstacles



(d) Average Computation Time per Iteration



(a) Number of Iterations for Target Achievement



(b) Average Total Path Length over 20 Random Maps

Figure 8: Comparative Analysis of MPC and APF Across Various Metrics